Defining Reuse

Michael Leyton

NASA ESDS WG Meeting 2008

Why this session?

At last ESDS meeting, October 2007, approximately once every five hours in the main sessions, someone would ask: “What is the difference between Use and Reuse?”, and a rather chaotic discussion would then begin to try to answer this question and usually fizzle out without clarity.

Also, at every few telecons, the exactly same question is asked, and a rather chaotic discussion follows, fizzling without without clarity.

Purpose of this session

Decide on a definition of *reuse* that clearly distinguishes it from *use*, so that an immediate quick answer can be given on such occasions.

Proposal

The crucial word that distinguishes *reuse* from *use* is the word:

Context

use switches to *reuse*, when the context changes.

i.e.,

use assumes the same context

reuse assumes a different context

Krueger's 5-fold taxonomy of processes that facilitate the reuse of a software artifact:

1. **Classification:** Organization of stored information in order to assist future queries and updates.
2. **Abstraction** Hides low-level details and implementation in order to assist understanding.
3. **Selection:** The agent who is constructing with reuse chooses what to reuse from the available reusable artifacts.
4. **Specialization:** Needed in white box reuse, where an artifact has to be altered to become reusable. In contrast, in black box reuse, an artifact is used without alteration.
5. **Integration:** Fitting the artifact being reused into the CONTEXT where it is going to operate.

Krueger, C.: Software reuse. ACM Computer Survey **24** (1992) 131–183

Notice #1-4 above are really servants of #5.

Therefore, the two crucial words are:

Context

Integration

They are clearly related:

Reuse is integration into another context.

Definition of Reuse:

The reuse of a software artifact is its integration into another context.

Notice: This explains and justifies the Reuse Readiness Levels developed by the working group:

Reuse readiness is the extent to which a software artifact is integratable into another context.

Also notice: this defines the difference between *use* and *reuse*.

Use does not involve integration into another context.

Definition of Reuse:

The reuse of a software artifact is its integration into another context.

Let us now consider what we mean by the term:

CONTEXT

Examples of a Context that can change:

1. Project
2. Machine architecture
3. Operating system
4. Software
5. Goal and requirements
6. Library
7. Needed run-time features.

Terms: Environment vs. Context.

In the literature, the word *environment* is used very frequently instead of *context*, although some major people use the word *context*.

However, *environment* is not as good a word as *context*.

The word *context* carries with it the sense of *integration* of an item within it, whereas the word *environment* does not.

For example, if you say “the context in which I was speaking” this explains why you said what you said. But if you say “the environment in which I speaking” this does not explain it. For example, you can say “I was discussing Einstein’s theory of relativity on the beach”. But the beach is not the explanation of why you were talking about Einstein’s theory of relativity; i.e., it is not the context. The context would be something like: “I was explaining to my son, who wants to study physics in school.” This is not the environment. The environment was the beach, but the context was my son’s career. So we should not use the word *environment*, we should use the word *context*.

Benefits of Reuse:

- **Reduced Costs**
- **Increased Quality**
- **Increased Productivity**
- **Reduced Risk**
- **Increased Performance**
- **Increased Interoperability**
- **Accelerated Development**

Final slide:
Therefore:
2-Sentence Statement about Reuse

The reuse of a software artifact is its integration into another context.
The purpose of reuse is to reduce cost, time, effort, and risk; and to increase productivity, quality, performance, and interoperability.